

Recherchebericht

Bearbeiter: Sebastian Dorn

Inhaltsverzeichnis

1	Begriffe	2
1.1	Apache Maven	2
1.2	Applet	2
1.3	Java-Applet	2
1.4	J2EE/Java	2
1.5	JSP	2
1.6	OLAT	2
1.7	Servlet	3
1.8	Spring (-Framework)	3
1.9	Tomcat	3
1.10	WAR-file	3
2	Konzepte	3
2.1	Client-Server-Modell	3
2.2	Framework	4
2.3	Apache-Lizenz	4
2.4	Open Source	4
2.5	Rollen-/Rechtekonzept	5
2.6	OLAT Schichtenmodell	5
3	Beschreibung der zu studierenden Aspekte	6
3.1	Zusammenspiel Tomcat, Spring und Brasato-Framework bei Anfragen an OLAT	6
3.2	Beschreibung der Konfigurationsdateien in <i>web-inf/classes/serviceconfig</i>	7
3.3	Vorgenommene Änderungen in der <i>olat.properties</i>	7
4	Quellen	7

1 Begriffe

1.1 Apache Maven

Maven ist ein Tool der Apache Software Foundation, welches auf Java basiert und ein fertiges Anwendungsprogramm erzeugt.

1.2 Applet

Applet setzt sich aus den Wörtern Application (Anwendung) und Snippet (Schnipsel) zusammen und bezeichnet:

- ein in einem Webbrowser laufendes Java-Programm, siehe Java-Applet
- ein Computerprogramm, das nicht als eigenständige Applikation betrieben wird

1.3 Java-Applet

Ein Java-Applet ist ein Computerprogramm, das in der Programmiersprache Java verfasst wurde und normalerweise in einem Webbrowser ausgeführt wird. Sie wurden eingeführt, um Programme in Webseiten ablaufen lassen zu können, die im Webbrowser (auf der Client-Seite) arbeiten und direkt mit dem Benutzer interagieren können, ohne Daten über die Leitung zum Server versenden zu müssen. Java-Applets waren Ende der 1990er Jahre mit ein Grund für den Erfolg und die schnelle Verbreitung von Java.

1.4 J2EE/Java

Java Platform, Enterprise Edition, abgekürzt Java EE oder früher J2EE, ist die Spezifikation einer Softwarearchitektur für die transaktionsbasierte Ausführung von in Java programmierten Anwendungen und insbesondere Web-Anwendungen. Sie ist eine der großen Plattformen, die um den Middleware-Markt kämpfen. Größter Konkurrent ist dabei die .NET-Plattform von Microsoft. In der Spezifikation werden Softwarekomponenten und Dienste definiert, die primär in der Programmiersprache Java erstellt werden. Die Spezifikation dient dazu, einen allgemein akzeptierten Rahmen zur Verfügung zu stellen, um auf dessen Basis aus modularen Komponenten verteilte, mehrschichtige Anwendungen entwickeln zu können. Klar definierte Schnittstellen zwischen den Komponenten und Containern sollen dafür sorgen, dass Softwarekomponenten unterschiedlicher Hersteller interoperabel sind, wenn sie sich an die Spezifikation halten, und dass die verteilte Anwendung gut skalierbar ist.

1.5 JSP

JavaServer Pages, ist eine von Sun Microsystems entwickelte, auf JHTML basierende Web-Programmiersprache zur einfachen dynamischen Erzeugung von HTML- und XML-Ausgaben eines Webservers.

1.6 OLAT

OLAT nahm seinen Anfang als studentisches Projekt an der Universität Zürich. Mit der Version 3.0 wurde das System im Jahr 2004 komplett neu entwickelt und ist seither als komponentenbasierte Java-Applikation Open Source erhältlich. Es läuft auf verschiedenen Systemen, wie Linux und Windows. Ab OLAT 7 wird die gesamte Applikation als WAR-file deployed und kann so einfach auf jeder Java Servlet Engine installiert werden.

1.7 Servlet

Als Servlets bezeichnet man Java-Klassen, deren Instanzen innerhalb eines Java-Webserver Anfragen von Clients entgegen nehmen und beantworten. Ist das Gegenstück vom Applet (Client) auf Serverseite. Weiterhin sind sie fester Bestandteil aller Java-EE-Anwendungsserver. Der Inhalt der Antworten kann dabei dynamisch, also im Moment der Anfrage, erstellt werden und muss nicht bereits statisch (etwa in Form einer HTML-Seite) für den Webserver verfügbar sein.

1.8 Spring (-Framework)

Das Spring-Framework ist ein quelloffenes Framework für die Java-Plattform. Ziel des Spring ist es, die Entwicklung mit Java/Java EE zu vereinfachen und gute Programmierpraktiken zu fördern. Spring bietet mit einem breiten Spektrum an Funktionalität eine ganzheitliche Lösung zur Entwicklung von Anwendungen und deren Geschäftslogiken.

Das Framework basiert auf folgenden Prinzipien:

- Dependency Injection: Den Objekten werden die abhängigen Objekte/Ressourcen zugewiesen
- Aspektorientierte Programmierung: getrenntes Entwickeln und Testen von verschiedenen logischen Aspekten einer Anwendung
- Templates: dienen dazu, die Arbeit mit einigen APIs zu vereinfachen, indem Ressourcen automatisch aufgeräumt werden und Fehlersituationen einheitlich behandelt werden.

1.9 Tomcat

Apache Tomcat stellt eine Umgebung zur Ausführung von Java-Code auf Webservern bereit. Dabei handelt es sich um einen in Java geschriebenen Servlet-Container, der JSP in Servlets übersetzen und ausführen kann. Dazu kommt ein kompletter HTTP-Server. Der HTTP-Server des Tomcat wird vor allem zur Entwicklung eingesetzt, während in Produktion zumeist ein Apache Web-Server vor den Tomcat geschaltet wird.

1.10 WAR-file

Web Archive oder Web Application Archive ist ein Dateiformat, das beschreibt, wie eine vollständige Webanwendung nach der Java-Servlet-Spezifikation in eine Datei im JAR- bzw. ZIP-Format verpackt wird.

2 Konzepte

2.1 Client-Server-Modell

Das Client-Server-Modell beschreibt eine Möglichkeit, Aufgaben und Dienstleistungen innerhalb eines Netzwerkes zu verteilen. Die Aufgaben werden von Programmen erledigt, die in Clients und Server unterteilt werden (s.Abb.1). Der Client kann auf Wunsch eine Aufgabe vom Server anfordern. Der Server, der sich auf dem gleichen oder einem beliebigen anderen Rechner im Netzwerk befindet, beantwortet die Anforderung. Die Regeln dieser Kommunikation, wie Format und Art der Informationen, werden durch Protokolle festgelegt. OLAT nutzt dieses Konzept.

Beispiele für Anwendungen sind: Datenbankserver, Webserver, Mailserver und Applikationsserver.

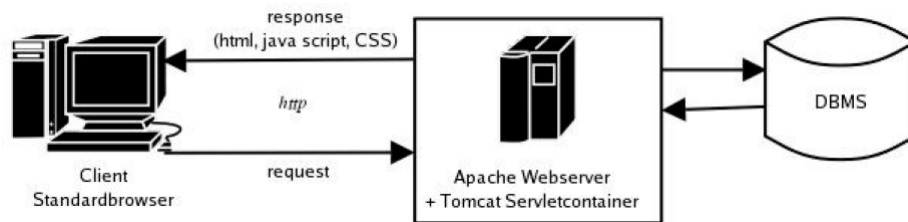


Abbildung 1: Client-Server-Modell

2.2 Framework

Ein Framework (englisch für „Rahmenstruktur“ oder „Fachwerk“) ist ein Programmiergerüst, das in der Softwaretechnik, insbesondere im Rahmen der objektorientierten Softwareentwicklung sowie bei komponentenbasierten Entwicklungsansätzen, verwendet wird. Ein Framework ist selbst noch kein fertiges Programm, sondern stellt den Rahmen, innerhalb dessen der Programmierer eine Anwendung erstellt, zur Verfügung, wobei u. a. durch die in dem Framework verwendeten Entwurfsmuster auch die Struktur der individuellen Anwendung beeinflusst wird. Ein Framework gibt somit in der Regel die Anwendungsarchitektur vor.

2.3 Apache-Lizenz

Die Apache-Lizenz ist die Freie-Software-Lizenz der Apache Software Foundation. Die aktuelle Version 2.0 wurde im Januar 2004 veröffentlicht. Die Apache-Lizenz wird von der Free Software Foundation als Lizenz für freie Software anerkannt.

Prinzipiell beinhaltet sie:

- Man darf Software unter dieser Lizenz frei in jedem Umfeld verwenden, modifizieren und verteilen.
- Wenn man sie verteilt, muss eindeutig darauf hingewiesen werden, welche Software unter der Apache-Lizenz verwendet wurde und dass diese vom Lizenzgeber stammt. Eine Kopie der Lizenz muss dem Paket beiliegen.
- Änderungen am Quellcode der unter der Apache-Lizenz stehenden Software brauchen nicht zum Lizenzgeber zurückgeschickt zu werden.
- Eigene Software, die unter Apache-Lizenz stehende Software verwendet, braucht nicht unter der Apache-Lizenz zu stehen.
- Die eigene Software darf nur dann Apache heißen, wenn eine schriftliche Genehmigung der Apache Foundation vorliegt.

Eigene Werke, die auf einem unter der Apache-Lizenz stehendem Originalwerk basieren, müssen:

- eine Kopie der Apache-Lizenz beinhalten bei modifizierten Dateien an auffälliger Stelle angeben, dass sie modifiziert sind
- alle Original-Urheberrechtsvermerke beibehalten
- falls das Originalwerk eine Textdatei namens „NOTICE“ enthält, die dort enthaltenen Urhebervermerke der verwendeten Dateien auf eine in der Lizenz genauer vorgeschriebene Art und Weise enthalten.

2.4 Open Source

Open Source und quelloffen ist eine Palette von Lizenzen für Software, deren Quelltext öffentlich zugänglich ist und durch die Lizenz Weiterentwicklungen fördert. OLAT ist eine Open Source Software und unterliegt damit den vorher aufgeführten Standards.

2.5 Rollen-/Rechtekonzept

Eine Benutzerrolle definiert Aufgaben, Eigenschaften und vor allem Rechte eines Benutzers in einer Software bzw. in einem Betriebssystem. Benutzerrollen werden verwendet um die Einstellungen der vorgenannten Bereiche nicht für jeden Nutzer einzeln festlegen zu müssen. Statt Benutzern Rechte direkt zuzuweisen, wird eine Benutzerrolle definiert, die dann vielen Benutzern zugeordnet werden kann. Dies erleichtert die Rechteverwaltung des Softwaresystems, da insbesondere bei Änderungen der Rechtestruktur nur die Rechte der Benutzerrolle angepasst werden müssen.

Rollen sind eine konzeptionelle Weiterentwicklung von Benutzergruppen. Ein Benutzer kann mehrere Rollen haben und seine Rechte ergeben sich dann durch die Vereinigung der Rechte aller Rollen. Moderne Softwaresysteme stellen dem Benutzer eine an seine Benutzerrollen angepasste grafische Benutzeroberfläche bereit.

OLAT stellt jedem Benutzer eine Anzahl von Rechten/Funktionen, abhängig von der Rolle in der sich der Benutzer befindet, zur Verfügung. Diese Rechte/Funktionen folgen einem Hierarchiekonzept, das vom einfachen Benutzer bis hin zum Administrator reicht. Einfachen Benutzern steht nur ein eingeschränkter Umfang der Funktionen, welche OLAT bereitstellt, zur Verfügung, während Administratoren den vollen Zugriff auf die angebotenen Möglichkeiten haben (s. Abb.2).

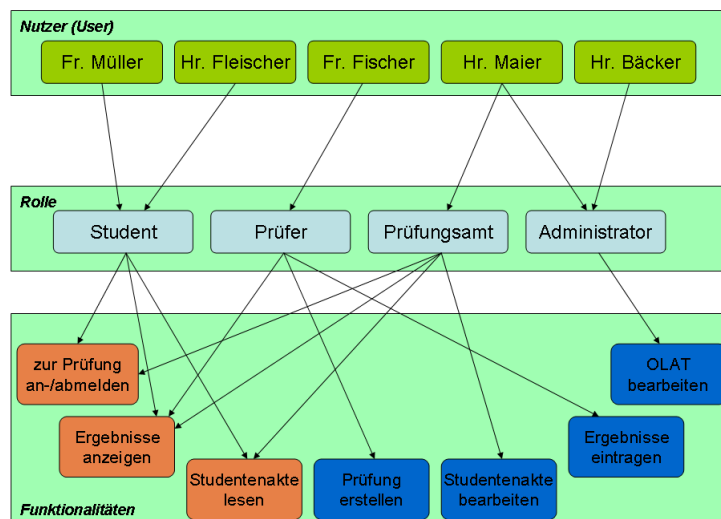


Abbildung 2: Auszug aus OLAT-PA Rollen-/Rechtekonzept

2.6 OLAT Schichtenmodell

OLAT ist eine modular aufgebaute Web-Applikation. Im klassischen Sinne hat es eine 5-Schicht-Architektur, geteilt in Client-Schicht, Präsentationsschicht, Business-Schicht, Integrations-Schicht und Ressourcen-Schicht (s. Abb.3). Jede Schicht ist in sich geschlossen und nur über definierte Schnittstellen ansprechbar.

Den Sockel jeder Webapplikation nach J2EE bildet die Ressourcen-Schicht, in dem alle zu speichernden Daten abgelegt werden. Die Integrations-Schicht kapselt diese Ressourcen und sorgt dafür das die Geschäftslogik in der Business-Schicht über eine einheitliche Schnittstelle darauf zugreifen kann. Die Präsentations-Schicht ist dafür zuständig, die von der Geschäftslogik übermittelten Inhalte in ein Format zu überführen, das der Client interpretieren und darstellen kann. Die Client-Schicht läuft auf dem Client im Webbrowser. Er enthält die sichtbare Applikation. Hier greift der User auf OLAT zu und ruft die verschiedenen Funktionen ab, die im GUI als Seiten sichtbar sind.

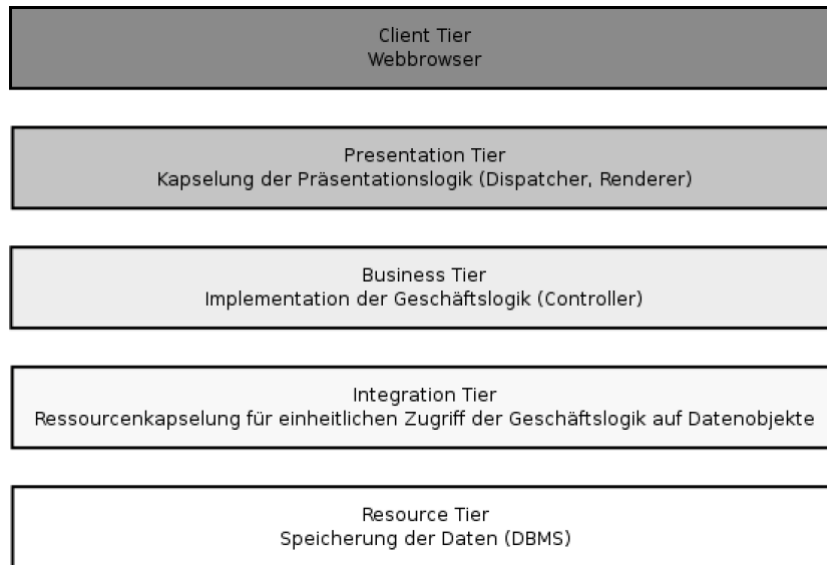


Abbildung 3: OLAT Schichtenmodell

3 Beschreibung der zu studierenden Aspekte

3.1 Zusammenspiel Tomcat, Spring und Brasato-Framework bei Anfragen an OLAT

Tomcat ist eine Middleware und stellt die Schnittstelle zwischen dem Nutzer mit seinen Webbrowser (Applet) und dem OLAT (Java-Server-Applikation).

Der Nutzer sendet eine *html*-Anfrage an das OLAT. Das OLAT selbst versteht aber nur Java-Anfragen. Der Tomcat-Server ist ein Java-Webserver, der entsprechende Anfragen unter Nutzung der Spring-API in Java umsetzt. Diese Anfrage kann nun vom OLAT (unter Nutzung des Brasato-Framework) bearbeitet werden. Die Antwort wird in umgekehrter Reihenfolge umgesetzt (s. Abb.4).

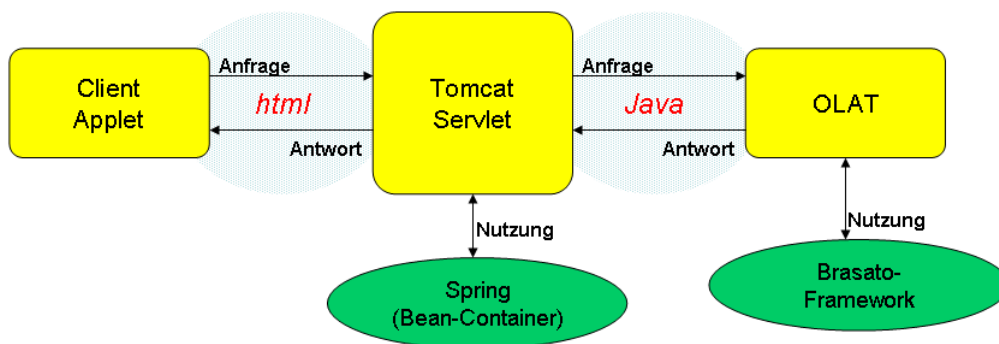


Abbildung 4: Zusammenspiel Tomcat, OLAT

3.2 Beschreibung der Konfigurationsdateien in *web-inf/classes/serviceconfig*

- brasatoconfig.xml*
- definiert die von AJAX akzeptierten Webbrowser
 - definiert die grafische Oberfläche des Systems
 - definiert die Konfigurationsparameter für den Tomcat-Server des OLAT
- olat.properties*
- Konfigurationsdatei für den Administrator zur Einrichtung der Systemstandardwerte im OLAT
 - Die Grundeinstellungen von *olat.properties* sind in verschiedenen Klassen von Spring integriert und wird vom Spring ausgelesen.
 - zur Einrichtung/Einstellung von:
 - Grundlegende Anwendungseinstellungen
 - SMTP (Mail)
 - Nutzer-LogIn/Registration
 - PlugIn-Konfiguration
 - Technische Einstellungen von OLAT
 - Webanwendungscontainer (z.B. Tomcat)
 - Datenbank
 - instant Messaging (z.B. Jabber)
 - Übersetzungswerkzeuge/-software
 - Volltextsuche
 - Sicherheit
 - Clustern

3.3 Vorgenommene Änderungen in der *olat.properties*

Alle folgenden Änderungen wurden in der *olat.local.properties* in der Tomcat-library (Tomcat/lib) durchgeführt. Die Änderungen können aber auch direkt in der *olat.properties* durchgeführt werden, da die geänderten Einstellungen in der *olat.local.properties* automatisch in die *olat.properties* übertragen werden. Hierbei wurden alle benötigten Werte auf *false* gesetzt.

Funktion	Einstellung
Deaktivierung des Gastzugang	<i>login.enableGuestLoginLinks= false</i>
Deaktivierung der Selbstregistrierung	<i>registration.enableSelfRegistration= false</i>
Deaktivierung des Macartney-Portlet	<i>portlet.macartney.enabled= false</i>
Deaktivierung der gui-demo-Tab	<i>guidemo.enabled= false</i>

4 Quellen

www.wikipedia.de
www.olat.org